

# 웹시스템설계 실습문서 Lab 08

최지현

[unidev@ajou.ac.kr](mailto:unidev@ajou.ac.kr)

이재현

[dlwogus8888@ajou.ac.kr](mailto:dlwogus8888@ajou.ac.kr)

2024-11-12

## 목차

<b>1 실습 과제</b>	<b>2</b>
1.1 프로젝트 디렉토리 구조 . . . . .	2
1.2 구현 요구사항 . . . . .	2
1.2.1 기본 요구사항 . . . . .	2
1.2.2 필수 기능 구현 . . . . .	2
1.2.3 컴포넌트별 상세 요구사항 . . . . .	3
1.2.4 데이터 구조 . . . . .	4
1.3 제출 방법 . . . . .	5
1.4 참고사항 . . . . .	5
1.5 Submission . . . . .	5
<b>2 Appendix</b>	<b>5</b>
2.1 Fetch data with React useEffect . . . . .	5

# 1 실습 과제

## 1.1 프로젝트 디렉토리 구조

---

```
src/
├── components/
│   ├── TodoInput.js
│   ├── TodoList.js
│   ├── TodoItem.js
│   └── TodoFilter.js
├── App.js
└── App.css
```

---

## 1.2 구현 요구사항

### 1.2.1 기본 요구사항

- React의 state, props, effect를 활용하여 Todo Application 구현

### 1.2.2 필수 기능 구현

#### 1. Todo 항목 기본 관리

- 우선순위 설정 기능
  - 사용자는 각 할 일에 대해 high, medium, low 중 하나의 우선순위를 필수적으로 지정해야 함
  - 우선순위는 시각적으로 구분되도록 표시 (high: 빨간색, medium: 노란색, low: 초록색)
  - 새로운 할 일 생성 시 기본 우선순위는 medium으로 자동 설정
  - 생성 후에도 우선순위 변경이 가능해야 함
- 마감일 관리
  - 각 할 일은 반드시 마감일(due date)을 포함해야 함
  - 마감일은 현재 날짜 이후로만 설정 가능하도록 제한
  - 마감일까지 남은 기간을 D-day 형식으로 표시 (예: 오늘로부터 3일 후 마감인 경우 'D-3')
  - 마감일이 지난 항목은 텍스트를 빨간색으로 강조하고 'D+n' 형식으로 표시
  - 마감 임박한 항목(3일 이내)은 경고 표시 추가
- 상태 필터링 시스템
  - 기본 상태 필터 (모든 할 일 / 진행중 / 완료됨) 제공
  - 우선순위 기반 다중 필터링 지원 (예: high와 medium만 표시)
  - 모든 필터 조건은 AND 로직으로 동작 (예: '진행중'이면서 'high' 우선순위인 항목만 표시)

- 필터 적용 시 실시간으로 결과가 업데이트되어야 함
- 현재 적용된 필터 조건을 시각적으로 명확하게 표시

## 2. 일괄 처리 기능

- 다중 선택 메커니즘
  - 각 할 일 항목 앞에 개별 선택용 체크박스 배치
  - 목록 상단에 '전체 선택' 체크박스 구현
  - 현재 필터링된 화면에 표시된 항목만 전체 선택 대상에 포함
  - 선택된 항목 수를 실시간으로 표시 (예: '3개 항목 선택됨')
  - 선택된 항목들은 시각적으로 구분되도록 하이라이트 처리
- 일괄 삭제 프로세스
  - 선택된 모든 항목을 동시에 삭제할 수 있는 버튼 제공
  - 삭제 전 확인 대화상자 표시 (예: '선택한 3개 항목을 삭제하시겠습니까?')
  - 삭제 완료 후 사용자에게 피드백 메시지 제공
- 일괄 상태 변경 기능
  - 선택된 항목들의 완료/미완료 상태를 일괄 변경하는 토글 버튼 제공
  - 상태 변경 시 확인 메시지 표시 (예: '선택한 항목들을 완료 처리하시겠습니까?')
  - 변경된 항목들은 애니메이션과 함께 새로운 상태를 반영
  - 일괄 상태 변경 후 적용된 필터에 따라 목록 자동 갱신
  - 작업 완료 시 성공 메시지 표시

### 1.2.3 컴포넌트별 상세 요구사항

#### TodoInput.js - 할 일 입력 컴포넌트

- 텍스트 입력
  - 빈 문자열 입력 시 등록 버튼 비활성화
  - 최대 50자 입력 제한
  - Enter 키로도 입력 가능
- 우선순위 선택
  - Select 박스로 우선순위 선택
  - 선택된 우선순위에 맞는 색상 미리보기
- 마감일 설정

- 오늘 이전 날짜 선택 불가

### TodoList.js & TodoItem.js - 할 일 목록 컴포넌트

- 할 일 목록 표시
  - 우선순위 높은 순으로 정렬
  - 같은 우선순위는 마감일 임박한 순
  - 완료된 항목은 목록 하단에 표시
- 항목별 기능
  - 체크박스로 완료 여부 토글
  - 완료 시 취소선 표시
  - 삭제 버튼으로 개별 항목 삭제
- 정보 표시
  - 우선순위를 색상으로 표시
  - 마감일까지 남은 기간 표시
  - 생성일자 표시

### TodoFilter.js - 필터링 컴포넌트

- 우선순위 필터
  - 체크박스로 다중 선택 가능
  - 선택된 우선순위에 해당하는 항목만 표시
- 상태 필터
  - 라디오 버튼으로 단일 선택
  - 전체/진행중/완료 중 선택

#### 1.2.4 데이터 구조

---

```
{  
  id: number,  
  text: string,  
  completed: boolean,  
  priority: 'high' | 'medium' | 'low',  
  dueDate: string,  
  createdAt: string  
}
```

---

### 1.3 제출 방법

- node\_modules 디렉토리를 제외하고 프로젝트 전체를 압축
- 압축 파일명: Lab08\_학번.zip

### 1.4 참고사항

- useEffect의 의존성 배열 적절히 설정
- Props는 필요한 데이터만 전달
- 컴포넌트 분리 시 재사용성 고려

### 1.5 Submission

제출 기한: 11월 12일 자정 / 지각 제출: 11월 14일 자정

## 2 Appendix

### 2.1 Fetch data with React useEffect

HackerNews 사이트 검색 API를 이용하여 데이터를 가져오는 예제이다. useEffect hook 함수에서 fetch를 이용해 백엔드 서버로부터 값을 가져온 뒤에 렌더링한다.

---

```
import React, { useState } from 'react';

function App() {
  const [data, setData] = useState({ hits: [] });

  useEffect(async () => {
    fetch('http://hn.algolia.com/api/v1/search?query=redux', {
      method: 'GET'
    }).then(response => response.json())
    .then(result => {
      setData(result.data);
    })
  }, []);

  return (
    <ul>
      {data.hits.map(item => (
        <li key={item.objectID}>
          <a href={item.url}>{item.title}</a>
        </li>
      ))}
    </ul>
  );
}
```

```
    );  
  }
```

```
export default App;
```

---

본 실습에서는 처음 렌더링 시 강의 목록 데이터를 받아와 화면에 표시할 수 있도록 아래의 코드를 그대로 사용한다.

---

```
useEffect(() => {  
  fetch('/api/courses')  
    .then(response => response.json())  
    .then(data => setCourses(data))  
    .catch(error => console.error('Error fetching courses:', error));  
}, []);
```

---